Computers in Spaceflight: The NASA Experience

- Chapter Six -
- Distributed Computing On Board Voyager and Galileo -

Voyager - The flying computer center

[**173**] After the cancellation of the Thermoelectric Outer Planet Spacecraft (TOPS) project as such, JPL proposed, and NASA funded, a project called Mariner-Jupiter-Saturn 1977. It was given the name [**174**] Voyager in the mid-1970s. Although TOPS' original mission was to conduct the Grand Tour of the four gas giant planets, Voyager was limited to flybys of the innermost two, Jupiter and Saturn. However, favorable gravity assists and hardware longevity made it possible to plan for a Uranus flyby by the Voyager 2 spacecraft and, potentially, a Neptune encounter. After visiting Jupiter and Saturn, Voyager 1 is to travel out of the plane of the planetary orbits and leave the solar system.

Voyager employs three dual-redundant computer systems per spacecraft. The first, the CCS, is nearly identical to that flown on Viking, performing sequencing and spacecraft health functions along with new ones necessitated by the addition of the other computers. Telemetry data formatting and transmission handled by the Flight Data System are done on Voyager with the help of a custom-built computer. Attitude control and articulation of the scan platform are accomplished with the third computer system. One concept from the STAR computer proposed for the TOPS, applicable to Voyager, is dormancy. JPL's project staff believed that equipment would last longer if unpowered[4]. Although both CCSs are always powered, rarely are both Flight Data Systems running, and both attitude control computers are never turned on at the same time. Full bit-for-bit redundancy is not maintained in the dual memories. For example, "expended" algorithms, such as the deployment sequence executed shortly after separation from the booster, need not be maintained[5]. Both memories are accessed by the single active processor in each system. The Flight Data System keeps a copy of its instructions in both memories, but intermediate data and variables can be stored in either memory. This seemingly casual attitude toward memory duplication tightens up considerably near encounter periods, which is one time that both CCS processors are in tandem mode.

Since there are three computer systems on Voyager, JPL had to establish another layer of organizational control over its flight hardware and software development. Whereas Viking was assigned a single Cognizant Software Engineer, Voyager had three, managed by a Spacecraft Software Engineer. H. Kent Frewing of JPL assumed this position in early 1974 and sent out a series of organizing memos during the first half of that year.** Frewing's February 20, 1974 note set out his duties and a project time line through the summer 1977 launch dates[6]. Manpower estimates for software development ranged from one programmer in 1974 and 1977, with a peak of four full-time programmers in late 1975. The small group allowed most work to be [**175**] done informally, easing communication. To provide some structure, Frewing established a Mariner-Jupiter-Saturn 1977 On-Board Software Design Team consisting of himself, Donald R. Johnson, the Flight Data System Cog Engineer, Stanley Lingon, the CCS Cog Engineer, and an Attitude and Articulation Control System representative[7]. They helped ensure the same close control of software development as on Viking, with good documentation and effective subroutine interfaces. The validation end of the software development process was handled by the Capability Demonstration Laboratory (CDL). Completed after the initial software was produced, it was a collection of either breadboard or flight surplus computer and science hardware, and its interfaces interconnected in the same way as those on the actual spacecraft. Its function is identical to that of the Shuttle Avionics Integration Laboratory (SAIL), in which both software and hardware changes could be tested to see if they functioned successfully[8]. Under this management umbrella, and with Cog Engineers constantly elucidating requirements from the science side and interpreting them to software engineers, each of the three computer systems took shape.

Voyager CCS: Parameters and Problems

NASA reeled from massive budget cuts during the 1970s. A changed political climate ended the Apollo era of near "carte blanche." Hampered by expensive Shuttle contracts as well as other factors, NASA management reduced its plans for unmanned exploration of the solar system. As Voyager developed under the new conditions, cost savings became a key ingredient in all engineering evaluations. JPL thus conducted a "CCS/CCS Memory Subsystem Design Inheritance Review" on January 17, 1974[9]. Held a year after Greenberg's proposal for standardizing the Viking computer, the Review resulted in the adoption of the Viking CCS as the Voyager CCS. The eventual hardware functional requirements document reads like a copy of the Viking document[10]. I/O interfaces with the new Flight Data System and Attitude Articulation and Control System computers are the major differences. Software such as the command decoder, certain fault processing routines, and others are fundamentally identical to Viking[11]. Here again, differences are related to the new computers. All command changes and memory loads for the other computers are routed through the CCS[12]. This required the addition of the routine MEMLOAD[13]. Another routine, AACSIN, was added to evaluate power codes sent from the Attitude Control computer as a "heartbeat" to inform the CCS of its health[14]. The frequency of the heartbeat, roughly 30 times per minute, caused concern [**176**] that the CCS would be worn out processing it. Mission Operations estimated that the CCS would have to be active 3% to 4% of the time, whereas the Viking Orbiter computer had trouble if it was more than 0.2% active[15]. As it turns out, this worry was unwarranted.

Part of the reason why the more complex Voyager spacecraft could be controlled by a computer with the same size memory as Viking is the ability to change software loads. In-flight reprogramming, begun when the programmable sequencers flew on Mariners, and brought to a state of high quality on Mariner X, was a nearly routine task by the time of Voyager's launch in 1977. Both the CCS and Flight Data System computer have been reprogrammed extensively. No less than 18 loads were uplinked to Voyager l during its Jupiter encounter. During long-duration cruise, such as between Saturn and Uranus, new loads are spaced to every 3 months[16]. As pioneered on Mariner X, a disaster backup sequence was stored in the Voyager 2 CCS memory for the Uranus encounter, and later for the Neptune encounter. Required because of the loss of redundancy after the primary radio receiver developed an internal short, the backup sequence will execute minimum experiment sequences and transmit data to earth; it occupies 20% of the 4K memory[17]. CCS programmers are studying ways to use some bit positions in a failed Flight Data System memory to compensate for the shortened memory in their system. A readout register in the Flight Data System has a failed bit, giving the impression that the entire memory has a one stored in that position in each word. Remaining "good" areas may be assigned to the use of the CCS[18].

Voyager Attitude Articulation and Control System Computer

JPL has been committed to three-axis stabilized spacecraft since it began designing probes in 1959. Attitude control systems maintain the proper pointing. The tasks assigned to the systems later expanded to include the actual operation of scanning platforms for imaging and other remote sensing instrument pointing. On the early Mariner missions the control systems consisted of analog circuits made up of hard-wired logic. By Mariner VIII, digital circuits replaced the analog electronics, and those were used on Mariner X as well as the

Viking Orbiter[19]. Viking's Lander used the Honeywell central computer to run its independent attitude control system[20]. A landing craft engaged in a powered descent needed far finer pointing than a spacecraft in free flight, and the bandwidth of a hard-wired system was insufficient to provide such control[21].

Future probes, however, might need computer-controlled attitude electronics due to complex mission requirements or unusual [**177**] spacecraft configurations. NASA's Office of Aeronautics and Space Technology funded a study of extended life attitude control systems as the TOPS project wound down in 1972. The result was a combination analog and digital programmable attitude control system. Dubbed "HYPACE," for Hybrid Programmable Attitude Control Electronics, it was a byte-serial processor with substantial power[22]. Using the same 4K, 18-bit-wide plated-wire memory from the Viking Orbiter computer, HYPACE added transistor-transistor logic (TTL) medium-scale integrated circuits to create a relatively fast (28-microsecond cycle) processor with index registers for addressing. Byte-serial architecture was possible because the TTL chips were designed for 4-bit parallel operation, so the 18-bit words could be moved around in five cycles instead of the 18 a serial machine would need, increasing overall speed. Index registering meant that the same block of code could be used for all three axes, reducing memory requirements. It appeared that the attitude control systems of future spacecraft would almost certainly benefit from such a computer.

Voyager was the first to do so, due to new requirements. One difference between Voyager and Mariner and Viking is that the latter two were fairly rigid in construction. Voyager's radioisotope thermoelectric generators, however, were mounted on a boom to keep radiation leakage away from scientific instruments. In addition, the magnetometer was boom mounted to avoid interference from spacecraft magnetic fields caused by motors, actuators, power buses, and electronics. Finally, the scan platform was also on a boom to give a better field of view. The extended booms made Voyager much less rigid in flight, with thruster firings and maneuvers causing the booms to flex, complicating the attitude control problem[23]. Additionally, the Titan III booster used for Voyager required a "kick stage" to successfully inject Voyager into the transfer orbit to Jupiter. Since the kick stage was kept simple, the spacecraft itself was required to do attitude control during firing, which entailed much narrower margins of control than the three-axis pointing in cruise[24].

JPL's Guidance and Control Section wanted to use a version of HYPACE as the computer for the Voyager. However, there was considerable pressure to build on the past and use existing equipment[25]. Greenberg proposed using the same Viking computer in all systems on the Voyager spacecraft that needed one[26]. A study showed that the attitude control system could use the CCS computer, but the Flight Data System could not due to high I/O requirements[27]. Wayne Kohl, the Viking computer Cog Engineer, thought that the Voyager project could save $300,000 by using the Viking machine for the attitude control function[28]. His division chief, John Scull, supported that idea, possibly because of budget pressure from NASA[29]. Raymond L. Heacock, as Spacecraft Systems Manager in the Voyager Project Office, and others from that organization were the key personnel [**178**] involved in making the final decision, influenced by the economy and feasibility of the idea[30]. Money could be saved in two ways by using the existing system: avoidance of new development costs and retraining of personnel.

Guidance and Control grudgingly accepted the CCS computer on the condition it be speeded up. Requirements for active control during the kick stage burn meant that real-time control programs would have to be written to operate within a 20-millisecond cycle, roughly three times faster than the command computer[31]. An executive for the attitude control computer differed in nature from those for either the command computer or the Flight Data System computer. Basically, the attitude control computer needed to run subprograms at different rates, requiring several cycles, as in Apollo, Skylab, and the Shuttle. Guidance and Control asked for a 1-megahertz clock speed but wound up getting about three quarters of that[32]. The attitude control engineers also added the index registers that proved so useful during the HYPACE experiment. Documentation for the system still refers to the attitude control computer as HYPACE, even though its heart was the command computer. General Electric, which built the command computer, naturally built HYPACE, but the rest of the attitude control system was constructed by Martin-Marietta Corporation in Denver.

Teoguer A. Almaguer was the hardware Cog Engineer for the attitude control computer, whereas H. Karl Bouvier led the software development group. Bouvier actually worked on an analysis team within the Guidance and Control Section, but the team members were afraid to use the word "software" in their name because their tasks might have been taken away and given to an existing software team in another division[33]. The programmers must have done an outstanding job, considering the slow processor and limited memory. At launch, only two words of free space remained in the 4K of plated wire[34]. Tight memory is now a problem because the scan platform actuators on Voyager 2 are nearly worn out, and software has to compensate for this during Uranus and Neptune encounter periods.

[**179**] Box 6-1: Voyager HYPACE Operation

HYPACE had four execution rates. Scan platform stepper motors and thruster actuators were among the routines executed during the 10-millisecond cycle. Attitude control laws and thruster logic executed in the 20-millisecond cycle. Scanning control and turn execution were placed in the 60-millisecond group, and the command interpreter and heartbeat were 240-millisecond routines[35]. In operation, the standard 10-millisecond time interrupt would cause all 10-millisecond routines to execute. It was time for one of the 20-, 60-, or 240-millisecond routines to run, it would be scheduled. Sometimes if the computer got too busy, the 240-millisecond cycle slipped to up to 350 milliseconds, but routines in that cycle were less critical than a routine to shut off an engine on time.

One thing needed on Voyager that did not exist when only single computers flew on unmanned spacecraft was an interface between the machines. The command computer could directly request data from either of its partners. A primary function of the command computer was to check periodically on the health of the other computers. Programmers in the Guidance and Control Section originally intended to send a "heartbeat" to the command computer each second[36]. This was later raised to once about every 2 seconds, partly because of the command computer overload problem mentioned above. To carry the heartbeat, six direct input lines, similar to the 3-bit synchronization bus on the Shuttle, ran from the HYPACE to the command computer. A "power code" was the content of the 6 bits transmitted on those lines. For example, power code 37 was the simple heartbeat. Others related to passing information such as pointing commands. Power code 66, called "the Omen," told the command computer to save disaster parameters, because a failure was imminent[37]. Every eight 240-millisecond cycles the heartbeat was sent. Between times, the attitude control computer conducted its self tests. If it failed, the heartbeat generator was bypassed. After about 10 seconds passed with no heartbeats, the command computer would issue a switch-over command to the backup processor.

A switch-over to the backup attitude control computer took place on Voyager 2 16 seconds after separation from the solid rocket stage[38]. Separation was so rough that the spacecraft was sent off attitude. Simultaneously, the booms were being deployed by the command computer. A thruster configuration initialization involving the plumbing for the thrusters delayed their acting to correct the attitude error.

[**180**] Since this was one of the mission-critical times that the command computer was turning in dual mode, the attitude control computer got two commands to initiate the plumbing. Executing the second

command pushed back the attitude control recovery even farther. Soon the computer exhausted its options and voluntarily stopped the heartbeat. When the backup came on-line it had no record of the gyro readings. Not knowing how bad things were was a blessing, as it executed a simple orientation and stopped the spacecraft roll[39]. Here is an instance where maintaining bit-for-bit identical memories would have been disastrous, as the backup computer would also have tied itself in knots.

Developing Voyager's Flight Data System Computer

Flight Data Systems handle the collection, formatting, and storage of science and engineering data on spacecraft. If the data are to be transmitted directly, a high rate of input and output is needed so that nothing is lost. If data transmission is deferred because a spacecraft is occulted from the tracking station, then the Flight Data System sends the data to a magnetic tape recorder known as the Data Storage System (DSS). As JPL progressed through Ranger to Surveyor to Mariner and to Viking, the rates of the data-handling requirements went steadily upward. This was because of increased instrumentation, greater sophistication in the spacecraft engineering systems, imaging equipment with better resolution (thus needing higher bit rates), and improved communications equipment permitting faster transmission of data. These changes led away from hard-wired Flight Data systems. One big step was the use of a digital memory on Viking to store different sequences of data handling. It was much like the microprogram in a central processor and for a similar purpose: to save hardware[40]. From there it was a short step to a full-fledged computer.

TOPS feasibility studies refer to a Measurement Processor Subsystem, the first time a separate computer was considered for flight data[41]. Although the command computer had been suggested as a possible Flight Data System machine, JPL engineers soon realized that even though the processing part of the job was well within the power of the computer, the I/O rates precluded its use.

JPL commissioned the development of a new computer from scratch and assigned Jack L. Wooddell to the job. Wooddell prepared an unusual document to tell the story of his work on the computer: a paper for a graduate computer science course taught by Dr. Melvin Breuer at the University of Southern California. Written around 1974, the paper includes what appears to be the flight version of the [**181**] design[42]. In it Wooddell lists the tasks he performed during the design period. He began by preparing a list of functions that the proposed Flight Data System was required to provide. These included sending control signals to sequence the science instruments, the ability to handle a wide variety of data rates and formats from the various instruments, potential for redesigning the mission in flight (as is now being done), monitoring engineering telemetry, and keeping to the reliability standard that no single failure result in loss of data from more than one scientific instrument or one-half the engineering sensors[43].

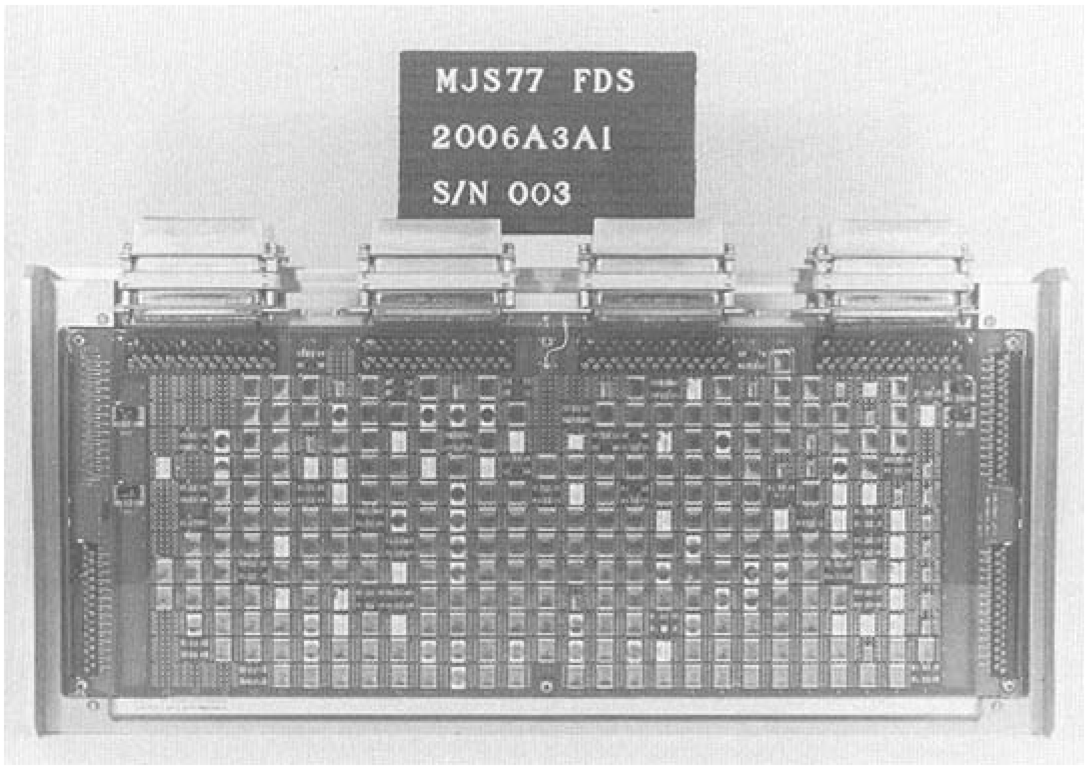

Figure 6-2. The Flight Data System hardware in its package. (JPL photo 360-751AC)

After determining requirements, Wooddell examined possible hardware and software tradeoffs. In an insightful memorandum, John Morecroft explained the concept of "soft logic" as a complement to the "hard logic" in the Flight Data System[44]. Writing in 1975, when the actual flight software began to be prepared, Morecroft pointed out that the program for the computer was actually a soft representation of hard-wired circuits. Conceptually, the memo stands as an explanation of the essential meaning of firmware in general. During the second phase of his work, Wooddell determined which functions could be handled by hardware and which should be left to the flexibility of [**182**] software. With those decisions made, a preliminary instruction set and logic design could be prepared.

Uniquely, Wooddell began working with a programmer in 1973, as soon as the instructions were ready[45]. Richard J. Rice of JPL began by developing software for a breadboard version of the data computer. The breadboard originally used the ubiquitous 4K memory of plated wire with 18-bit words and 150 of the same low-power TTL ICs used in other JPL machines[46]. Instruction execution times for this version ranged from 12 to 24 microseconds. Rice's prototype flight program, developed on the basis of what was then known about Voyager instrumentation and previous experience, showed that the processor speed should be doubled[47].

Two significant hardware changes solved this problem. One hardware modification added direct memory access circuits and provided for using them on each instruction cycle. Direct memory access capability meant that some data could be sent directly to the memory without having to go through the central processor. In other computers, direct memory access is permitted as a sort of interrupt and is often referred to as "cycle stealing" because it takes time away from instruction execution. In the data computer, it would have been foolhardy to do direct memory access in that way because the data rate was so high that the instructions might never get a chance to be executed quickly enough for time-critical sequencing. Wooddell solved this by adding a direct memory access cycle to those instructions that did not already have cycles in which the memory was accessed[48]. By adding that cycle all the instructions took the same time to execute regardless of direct memory access, making it easier to predict program run times and to guarantee the memory access rate[49]. Rice, who suggested the change, later said that his programming job would have been impossible without it[50].

The second hardware modification to Voyager's data computer led to a first in spaceflight computing: volatile memory. After the first round of prototype programs, an intermediate hardware design evolved using CMOS ICs[51]. This type of circuit is very low powered, fast, and can tolerate a wide range of voltages, making it excellent for space use. Early in the 1970s, CMOS was still relatively new, so it was with some risk that JPL chose the circuits. To go along with the new CMOS processor, the data computer group fought for CMOS memories as well. Trying to drive a slow plated-wire memory with fast CMOS circuits would have negated the attempt to speed up the computer. However, CMOS memories are volatile, in that if power is cut off, the data stored in them disappear. The designers of previous manned and unmanned spacecraft avoided volatile memories, fearing that power transients would destroy the memories at critical mission times. Voyager management had to be convinced that the risk was acceptable.

[**183**] James T. Kinsey, a JPL manager, was instrumental in getting the semiconductor memory accepted because a method of providing backup power was devised[52]. Voyager's primary electricity is alternating current. The radioisotope generators produce direct current, which is converted. By running a separate power line from the direct current bus fed by the generators to the CMOS memories, the only way power would be lost is if a major catastrophe destroyed the generators. If that happened there would not be any need for a data computer anyway. Enough voltage is supplied to retain the information in memory and in the registers in the processor that contain the state vector[53]. Success with the CMOS memory led to the adoption of all CMOS circuits in both computer systems on the Galileo spacecraft. Along with the new chips, the memory changed with an expansion to 8K. Two "external" address bits were added to flag whether the top or bottom half of the memory is being accessed[54]. One bit is used to select the memory half used for data access; the other, for the half used for instruction access.

Eventually, the cycle of prototyping and interaction between Rice and Wooddell stopped as a final design was accepted. Wooddell wrote that the extensive use of breadboards instead of paper designs optimized the process[55]. His method, although not strictly "software first" was certainly software sensitive. Martin-Marietta's experiences with a software first philosophy as described in the previous chapter indicate that Wooddell had a clearer idea of his objective than did Martin. The job done on the Flight Data Systems computer is a good model of fine engineering practice in developing a total system.

Voyager Flight Data System Software

The original software development for the data computer has essentially been a two-man show since 1975, beginning when Edgar M. Blizzard joined Richard Rice to develop the flight version of the code. Others have been involved in testing and management, but these two JPL engineers have been the key programmers for the entire mission to date. They sit in the same area as the "Laboratory Test Set," an Interdata computer and peripherals that contain the software simulator of the data computer and the assembler and flight load generator. Across from them is the CDL, the loose conglomeration of hardware that represents the real spacecraft. From start to validation to release, their tools were within sight, and certainly hearing, since the room is filled with the constant hum of spinning disks, occasional clattering printers, and the undefinable sound of computers crunching numbers.

Rice characterized the unique nature of the data computer software this way: "We didn't worry about top-down or structured;....

[**184**] Box 6-2: Voyager Flight Data System Computer Architecture

Voyager's data computer is different from most small general-purpose computers in several ways. Its special registers are kept in memory, permitting a large number (128) of them. Wooddell also wrote more powerful shift and rotate instructions because of data-handling requirements. Despite its I/O rate, the arithmetic rate is quite slow, mostly due to byte-serial operation. This meant 4-bit bytes are operated on in sequence. Since the word size of the machine is 16 bits, it takes six cycles to do an add, including housekeeping cycles[56]. If all the arithmetic, logic and shifting were not done in the general registers, the machine would have been even slower. Reflecting its role, in addition to the usual ADD, SUB, AND, OR, and XOR instructions found on most computers, the data computer has many incrementing, decrementing, and machines instructions among the 36 defined for the flight version of the machine [57].

Overall, the Flight Data System requires 14 watts of power and weighs 16.3 kilograms[58]. Its computer needs just one third of a watt and 10 volts, less than the power required for a temperature sensor[59]! At first the estimated throughput required was 20,000 16-bit words per seconds[60]. By flight time, the instruction execution rate was 80,000 per second, with data rates of 115,000 bits per second, much higher than previous Flight Data Systems[61] . The dual processor/dual memory architecture of the command computer and attitude control computer is repeated in the data computer. There was no provision for automatic switch-over in case of failure. A command from the ground routed by the command computer is necessary for reconfiguration[62]. Note that the attitude control computer can be switched by the command computer without ground intervention because it is much more critical to retain orientation.

....we just defined functions"[63]. One important function is the software's provision of basic timing for the entire spacecraft, not just itself. It is also required to provide the capability to read out the memories of all three computers, under orders of the command computer[64]. Don Johnson, the Cog Engineer, determined other requirements and interfaces with the scientific instruments. Rice called him "Mr. PDS," claiming that Johnson often knew more about the scientific instruments than the scientists themselves: "If someone forgot something, Johnson knew it"[65]. Raymond L. Heacock, Voyager Project Manager, said that Johnson was largely responsible for the overall success of the system, including the design[66]. Rice said that Johnson's ebullient style and competence worked well in the informal mode in which the data computer requirements were set, which was a fully iterative process. New software needs continued to be discovered during the mission, which is one reason why a programmable machine [**185**] was chosen. For example, at one point Rice and Blizzard were asked to create software to determine where the limbs of satellites were so that imaging could be started[67]. Development of some programs was deferred until after launch, such as the Saturn encounter program, when better data on the telecommunications rates and specific science requirements would be available[68].

Allowing for constant change mandated certain controls over the data computer's memory. A limit of 90% capacity was set in 1976 by Frewing, the Software Cog Engineer[69]. Though later abandoned, the constraint indicated the software management's early concern about memory overruns. Also, since the machine can directly address the lower 4K of memory, programs were to be kept there, with the upper portion for transient data[70]. Later, the flight configuration of the computer evolved to one processor accessing both memories. Therefore, a copy of the programs is kept in the lower portion of each memory, but both upper portions are usable by the single processor as a scratch pad[71]. If dual mode is required, the memories are separated. Experience has produced increased confidence in the memories. At first, complete loads had to be sent when an update was done; recently, pieces of software have been allowed to be inserted in the programs. Full redundancy between the memories is not now automatically maintained[72].

Box 6-3: Flight Data System Computer Executive

Like the command computer, the data computer has a simple executive. Time is divided into twenty-four 2.5-millisecond intervals, called "P periods." Each 24 P periods represent one imaging system scan line. Eight hundred of those lines is a frame. At the beginning of each P period, the software automatically returns to memory location 0000, where it executes a routine that determines what functions to perform during that P period[73]. Care is taken that the software completes all pending processes in the 2.5-millisecond period, a job made easier by the standardization of execution times once the direct memory access cycle was added.

Voyager's Future

Voyager software development continued into the late 1980s. Kohl, Wooddell, Greensburg, Deese, Johnson, Kopf, and others closely connected with the hardware of Voyager's computers were then on other projects, but Rice and Blizzard and their counterparts on the command computer and attitude control computer were still programming, [**186**] preparing Voyager 2 for Uranus and Voyager 1 to discover the boundary of the solar wind. An increasing problem as the spacecraft recede from the earth is the reduction in the data transmission rate. The closer a spacecraft is to earth, the higher the bandwidth possible. Computer loads that once took minutes now take hours because error checking by retransmitting to earth is slowed. In the summer of 1984, a Flight Data System software load took 4 hours, and the situation cannot improve[74]. Voyager Project officials decided to use the Flight Data System in dual processor mode for the first time for the Uranus encounter to provide image data compression. Thus, the information content remained high even though the transmission rate was grossly reduced[75].

Voyager's computer system did not carry on to the next JPL project. Galileo combined the CCS and the Flight Data System into a single Command and Data System. This is logical from JPL's standpoint because both systems are the responsibility of the same Information Systems Division. Attitude control is provided by a separate computer. Whereas Voyager was a functionally distributed system with dual redundancy, Galileo's Command and Data System contains computers that do true distributed processing and use a new concept of redundancy. That system may be a model for the future, as it can impact designs aimed at complex spacecraft with extensive data processing needs, such as the Space Station and Mariner Mark II, both due in the 1990s.

** He was replaced in early 1976 by Chnstopher P. Jones, who designed the integrated fault protection algorithms used on the mission, but Frewing laid the groundwork for management of the software.
